**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1-29. (Canceled)

30. (Previously Presented) A system for parsing a web-document based on elements, which is provided to an application of a handheld terminal when the system calls the web-document to provide it to the handheld terminal, comprising:

a word parser that separates a token on the basis of markup and non-markup by referring to a token table for all markup data necessary for a kind of document to be supported, wherein a same string of the web-document provided to the terminal has a different token according to whether it is a markup or a non-markup; and

a syntax parser that parses a contents model on the basis of a document type definition (DTD) of each document, parses each syntax on the basis of the result of parsing the contents model, and generates a tree-based object on the basis of graphic user interface (GUI) of the terminal.

31. (Previously Presented) The system of claim 30, wherein the word parser comprises:

a comment parser that processes a comment and a space;

a markup start parser that recognizes a markup start tag and generates a token;

an attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token, wherein the attribute parser is configured to recognize a name of an attribute or to recognize a value of the attribute.

32. (Previously Presented) The system of claim 30, wherein the syntax parser comprises:

an XML verifier that verifies whether a corresponding document is composed suitable for each DTD on the basis of the token generated by the word parser; and

a terminal GUI-based object generator that matches the analyzed markup and a GUI of the terminal.

33. (Previously Presented) The system of claim 30, wherein the parsing system integrally parses a web-document composed on the basis of any one of SGML and XML related to HTML, XHTML, mHTML, cHTML, WML and HDML.

34. (Previously Presented) The system of claim 30, wherein the parsing system can be applied to any handheld terminal and select a kind of an element to be parsed according to a specification of each of the terminals.

35. (Previously Presented) A method for parsing a called web-document of a web-server, the method comprising:

reading a token from the web-document and parsing the token;

if the token is not a defined start tag or if the token is a comment or a space as result of the reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

as the result of the parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears, terminating, and if the end tag corresponding to the defined start tag does not appear, ignoring and returning.

36. (Currently Amended) The method of claim 35, wherein the parsing comprises:

if the read token does not include ~~a~~ the defined start tag, reading the data continuously until the end tag appears, and if the end tag corresponding to the defined start tag does not appear, ignoring the token; and

reading a new token.

37. (Previously Presented) A recording medium for storing a program for parsing a called web-document of a web-server, the recording medium being read by a computer, the program comprising the functions of:

reading a token from the web-document and parsing the token;

if the token is not a defined start tag or if the token is a comment or a space as a result of the reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

parsing the attribute of the element from the token, storing GUI-related information of the element, and parsing contents of the element;

if the contents of the element are parsed character data as result of the parsing, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears; and

in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears, terminating, and if the end tag does not appear, ignoring and returning.

38. (Previously Presented) A system for parsing a web-document based on elements to provide contents thereof to a handheld terminal, comprising:

a word parser that extracts and separates tokens representing the web-document supplied regardless of a kind of a markup language used to compose the web-document by referring to a token table, wherein a same string of the web-document provided to the terminal has a different token according to whether it is a markup or a non-markup; and

a syntax parser that parses syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generates an object on the basis of GUI of the terminal.

39. (Previously Presented) The system of claim 38, wherein the token table comprises:

tokens defined in an XML document;

keywords defined in DTD for all documents provided to the handheld terminal; and

a list of elements which can be supported by each terminal.

40. (Previously Presented) The system of claim 38, wherein the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

an attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

41. (Previously Presented) The system of claim 38, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

42. (Previously Presented) The system of claim 38, wherein the contents model means a hierarchy of elements and an attribute list, and is defined in a DTD for all documents provided to the handheld terminal.

43. (Previously Presented) The system of claim 38, wherein the syntax parser comprises:

an XML verifier that verifies whether a web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and a GUI of the terminal.

44. (Currently Amended) A system for parsing a web-document based on elements, comprising:

a token table comprising tokens defined in an XML document, keywords defined in a document type definition (DTD) for documents provided to a handheld terminal, and a list of elements, which can be supported by each terminal;

a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute;

a contents model determined by DTDs for the documents provided to the terminal that includes a hierarchy of elements and an attribute list; and

a syntax parser that parses syntax for the tokens extracted and separated by the word parser on the basis of the contents model, and generates an object on the basis of GUI of the terminal through the parsed syntax.

45. (Previously Presented) The system of claim 44, the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

the attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

46. (Previously Presented) The system of claim 44, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

47. (Previously Presented) The system of claim 44, wherein the syntax parser comprises:

an XML verifier that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and a GUI of the terminal.

48. (Previously Presented) A handheld terminal, comprising:

an integral parser that parses a web-document composed of a predetermined markup language supplied from a web-server;

a memory that stores information parsed by the integral parser; and

an application program using information extracted from the integral parser, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute.

49. (Previously Presented) A handheld terminal comprising an antenna, a CPU, a peripheral circuit, a vocoder, a memory, and an audio codec, wherein the memory comprises:

an integral parser that calls a web-document supplied from a web-server regardless of a markup language used to compose the web-document and parses the web-document on the basis of elements, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute; and

an application program using information extracted from the integral parser.

50. (Previously Presented) The handheld terminal of claim 48, wherein the integral parser comprises:

a token table comprising tokens defined in an XML document, keywords defined in a DTD for all documents provided to the handheld terminal, and a list of elements, which can be supported by each of the handheld terminals;

the word parser that extracts and separates all tokens of the document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to a token table;

a contents model defined in the DTD for all documents provided to the terminal and meaning a hierarchy of the elements and an attribute list; and

a syntax parser that parses syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generates an object on the basis of GUI of the terminal through the parsed syntax.

51. (Previously Presented) The system of claim 50, wherein the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

an attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

52. (Previously Presented) The system of claim 50, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied document character by character, recognizes a token of the document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

53. (Previously Presented) The system of claim 50, wherein the syntax parser comprises:

an XML verifier that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and a GUI of the terminal.

54. (Previously Presented) The handheld terminal of claim 48, wherein the application program comprises an object based on a GUI of the handheld terminal.

55. (Previously Presented) A method for parsing a web-document supplied from a web-server, the web-document being composed of a predetermined markup language, the method comprising:

reading a token from the web-document by referring to a token table, extracting and separating the token;

if the extracted and separated token is not a defined start tag or if the token is a comment or a space, ignoring the token;

when the extracted and separated token is recognized as the defined start tag, parsing an attribute of an element from the token and storing GUI-related information of the element;

parsing contents of the element after parsing the attribute of the element;

as the result of the contents parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, determining whether an end tag appears;

as the result of the previous step, if the end tag does not appear, repeating the above steps, and if the end tag appears, determining whether the end tag corresponds to the defined start tag; and

as the result of the previous step, if the end tag corresponds to the defined start tag, terminating, and otherwise, ignoring and returning.

56. (Previously Presented) The method of claim 55, wherein the attribute parsing comprises:

if the extracted and separated token does not include a defined start tag, reading the data continuously until the end tag appears, thereby ignoring the token; and

reading a new token.

57. (Previously Presented) A handheld terminal, comprising:

an integral parser that parses web-documents composed of a plurality of predetermined markup languages on the basis of elements, wherein the integral parser includes a

word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute;

a memory that stores information parsed by the integral parser; and

an application program using information extracted from the integral parser.

58. (Previously Presented) A system, comprising:

a content provider configured to provide first type documents using a first markup language and second type documents using a second markup language different from the first markup language; and

a handheld terminal that receives the first and second type documents from the content provider, wherein the handheld terminal comprises:

an integral parser configured to parse both a first type document and a second type document on the basis of elements to extract information thereof, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, and wherein the word parser includes an attribute

parser configured to recognize at least one of a name of an attribute or a value of the attribute; and

an application program configured to receive the information extracted from the integral parser.

59. (Previously Presented) The system of claim 51, wherein the attribute parser includes a first attribute parser configured to recognize a name of an attribute and a second attribute parser configured to recognize a value of the attribute.

60. (Previously Presented) The system of claim 59, wherein if the value of the attribute is a keyword, the first attribute parser recognizes the name and the value of the attribute at once without distinguishing the name from the value.

61. (Previously Presented) The system of claim 30, wherein the different tokens are as follows: <html> represents an element type, html represents parsed character data, <!--html--> represents a comment.